

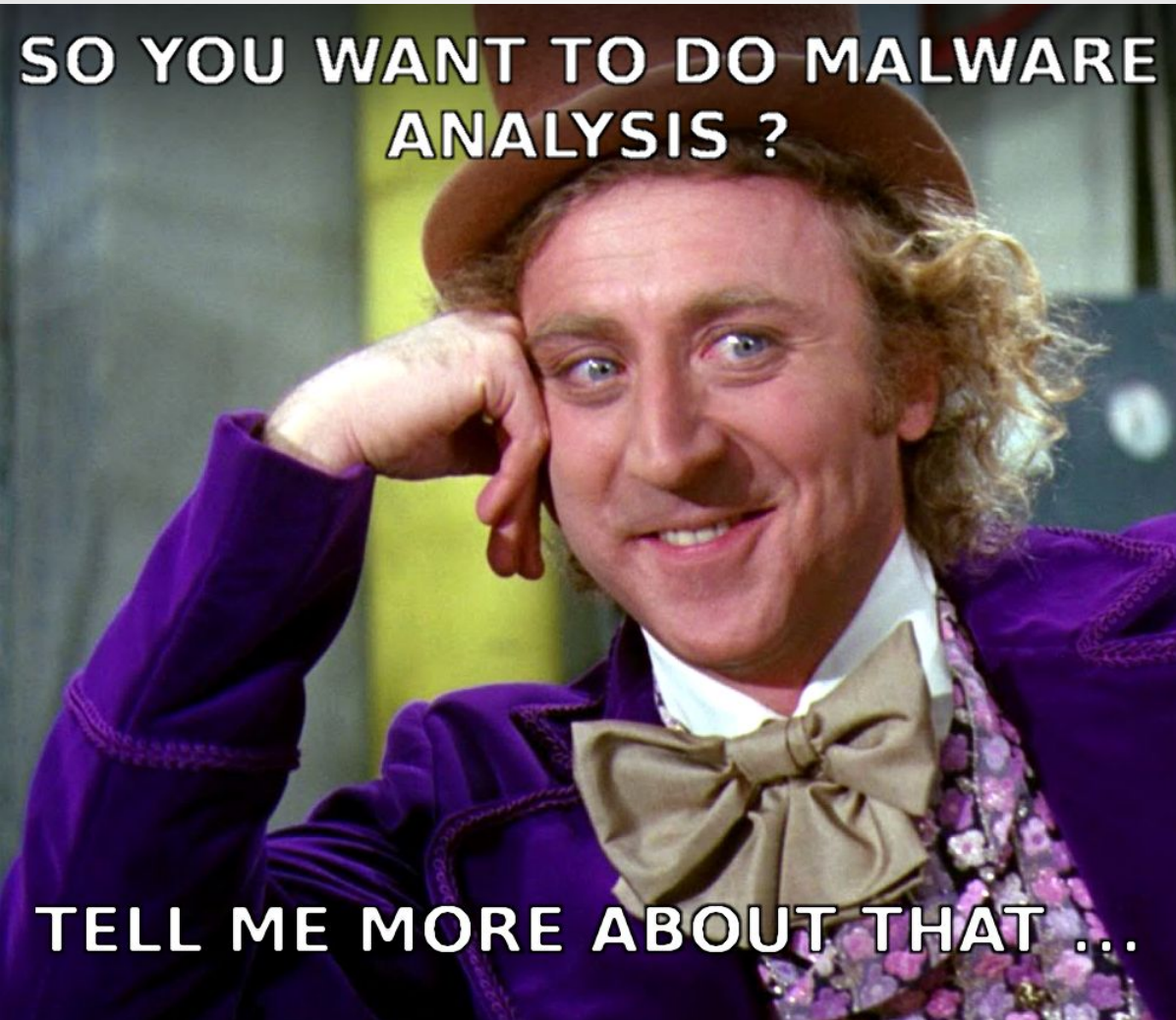
# Do Machines Dream of Binary Files



Marwan Burelle - LSE Summer Week 2016

We want to classify binary files.

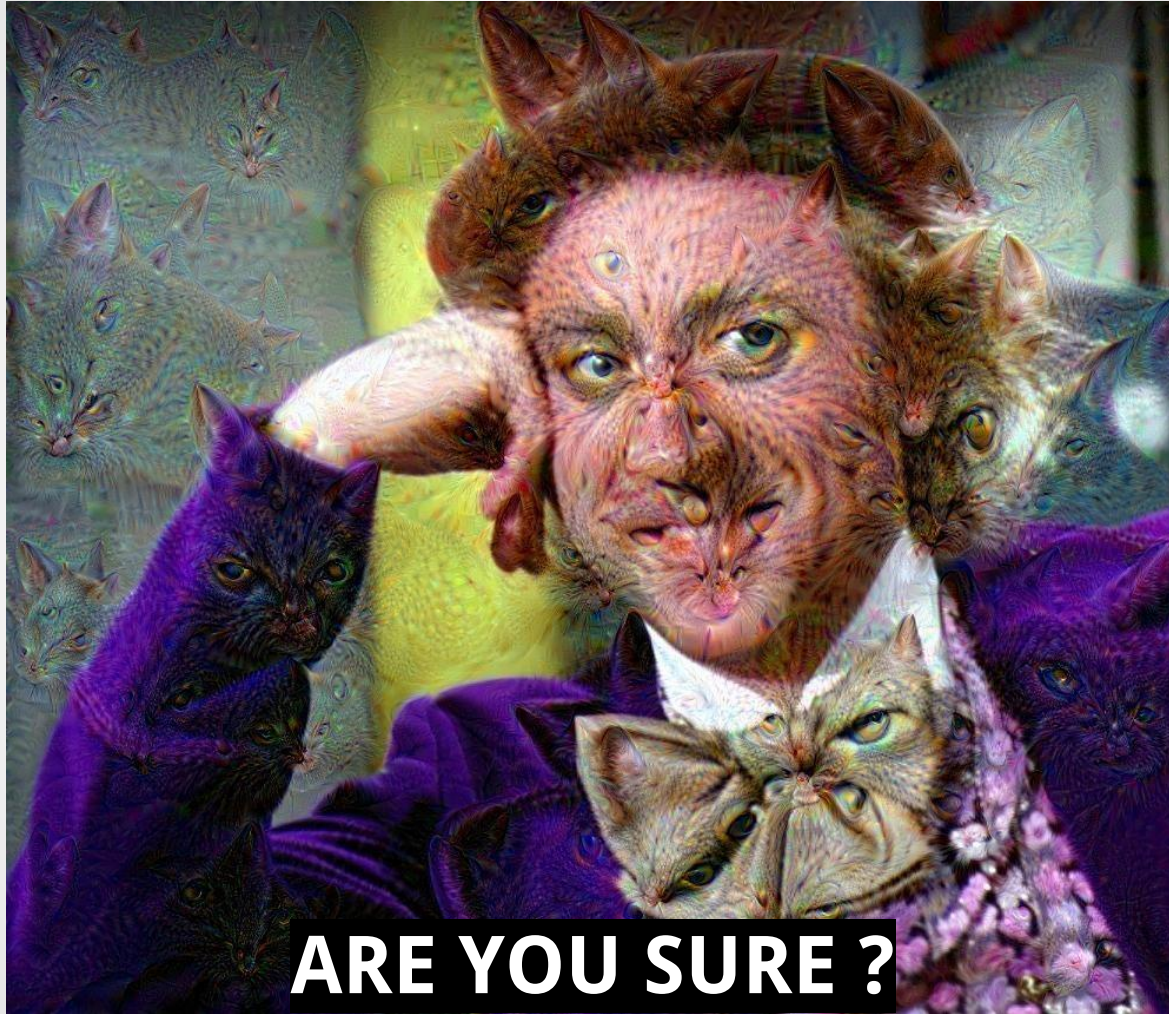




**SO YOU WANT TO DO MALWARE  
ANALYSIS ?**

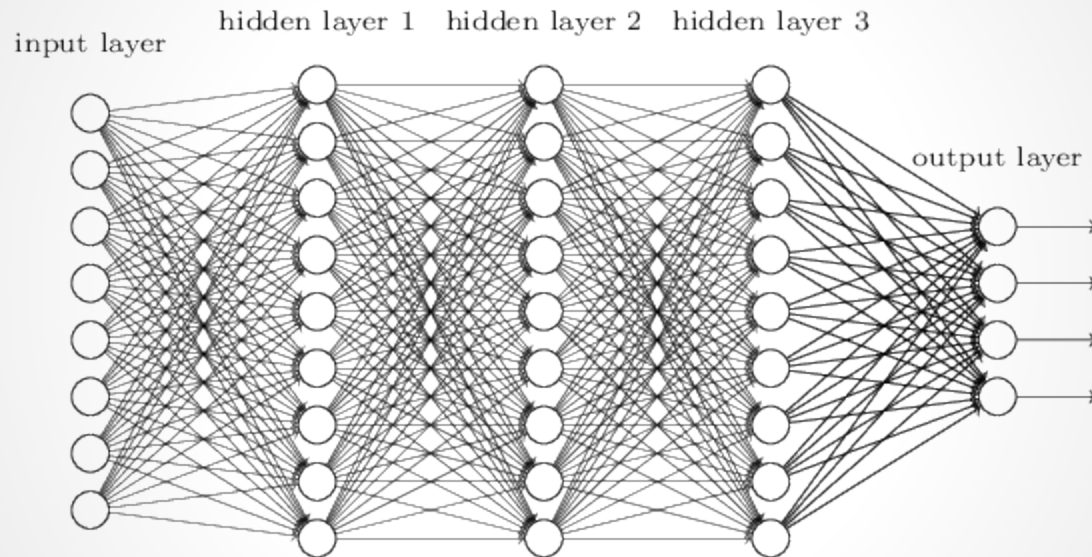
**TELL ME MORE ABOUT THAT ...**

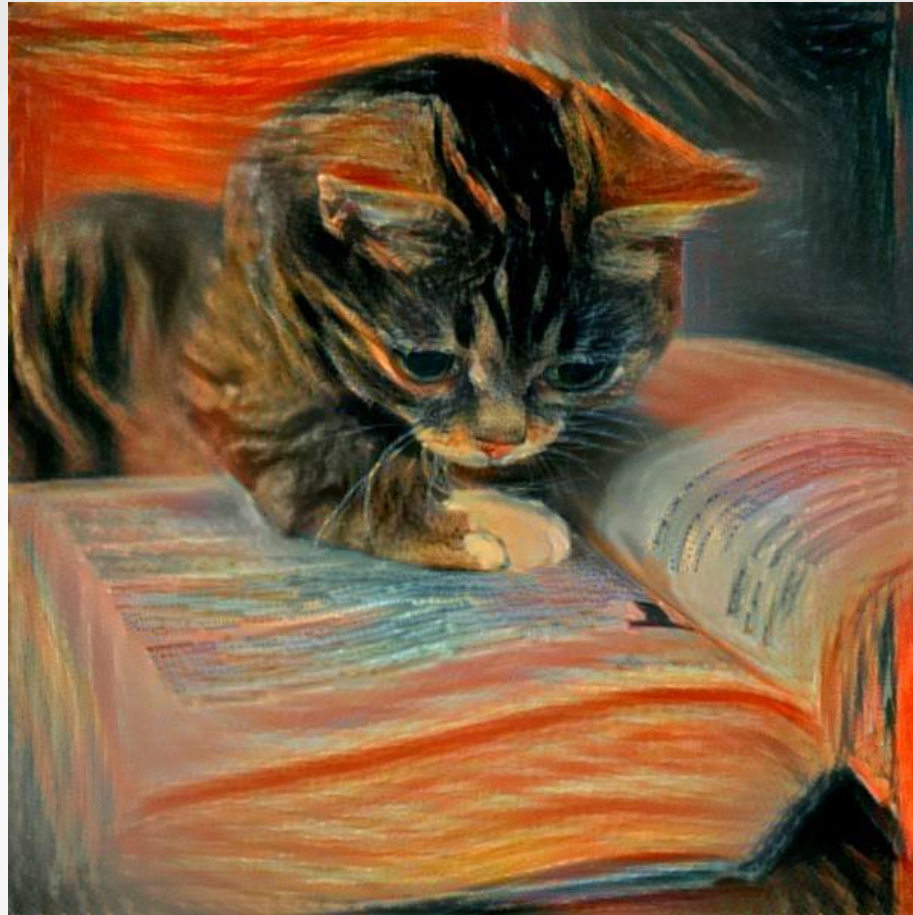
**Let's try deep learning ...**



**ARE YOU SURE ?**

# Deep Neural Networks



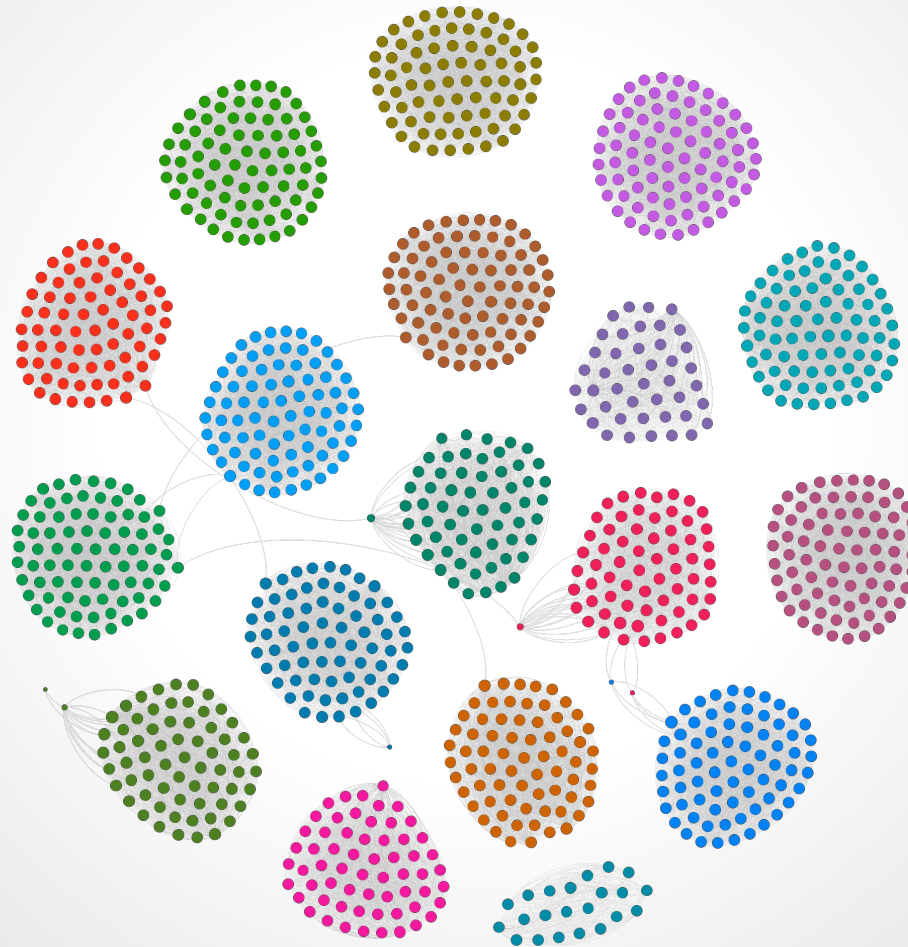


# Learning Distances

**Can I build a graph using a deep NN ?**



# Graph Clustering



# Problem

Entity: vector of *features*

Goal: a graph of entities

Distance: the cost on edges

- How can we compute distance ?
- Which edges should we keep ?

# First pass

- Classic distances don't work
- Weighted means, a little better but not enough
- Our problem is non-linear !

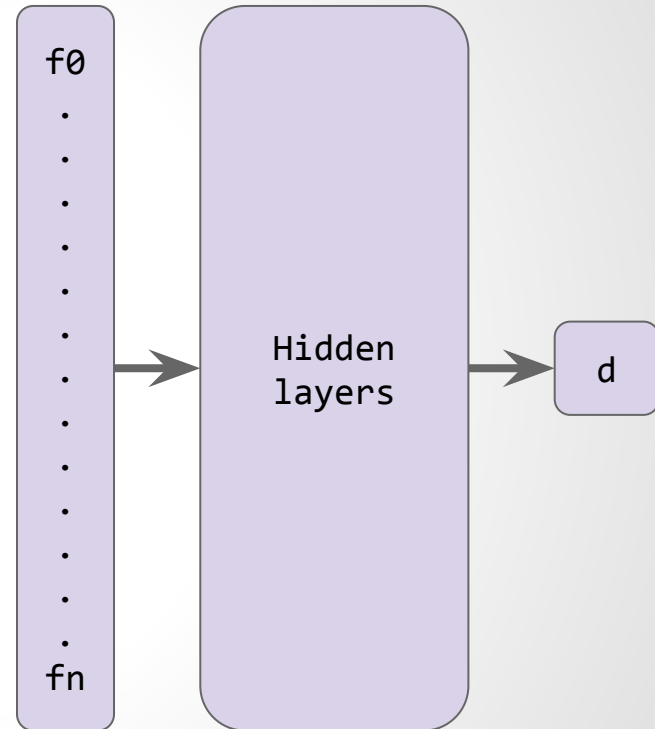
# Pre-labeled data and non-linear function ?



**Use Deep-Learning !**

# A NN for distance

- Regression problem
- Reference value for  $d$ :
  - same class:  $0$
  - different class:  $1$
- Deep network:
  - 3 Dense layer
  - ReLu activation
  - mean squared error
  - Adam optimizer
  - use dropout

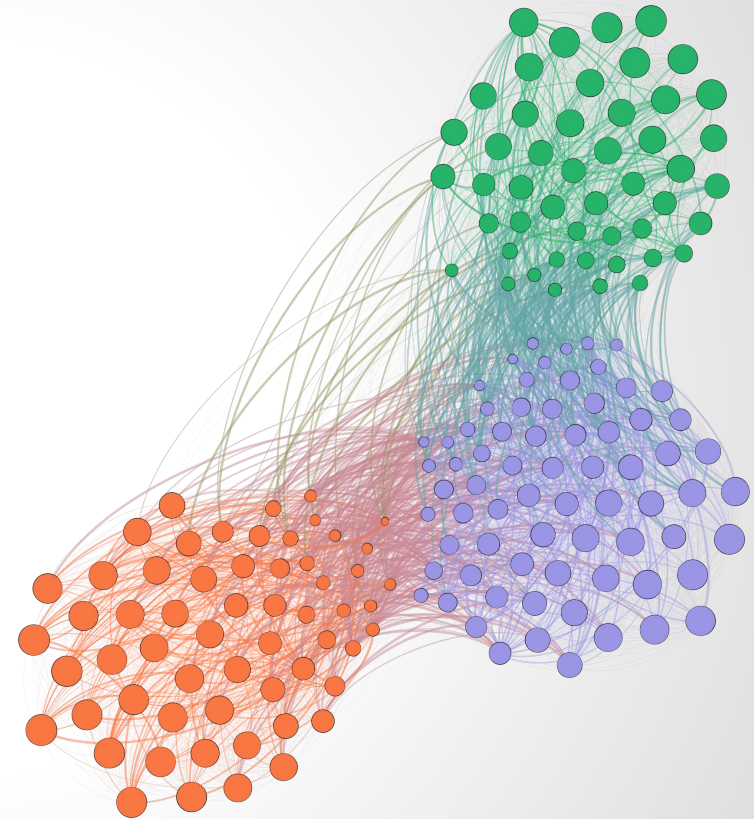


# Cut value

- Clustering needs sparse graph
- Which edges should we kept ?
  - Compute a *cut-value*
  - Remove edges with higher cost
- Good cut value ?
  - mean, median ... not really accurate
  - mean of means:
    - compute means of in-class and out-class edges
    - use mean of these two means
    - yields good results

# Example: wine data

- Wine chemical data
- 3 classes
- 178 samples
- 13 features
- tailored for ML testing



# Example: labeled malware

- Training dataset
- Extracted from 10868 files
- 9 classes (malware families)
- 9 Features
  - basic
- 30% rare





# Remaining Issues

- Need to generate all possible edges
  - on malware samples: 118,113,424 edges (13GB) !
  - solution ? stream samples, work on subset ...



- Features are important too
  - features used for malware were not accurate
  - solution ? better features extraction ...

**Can we use deep learning to extract features ?**



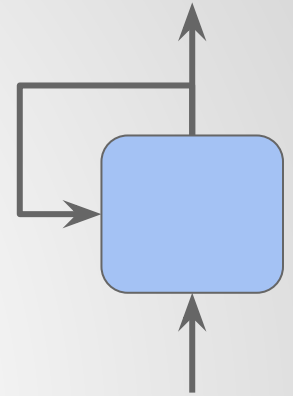
I want to see that ...

# Recurrent NN

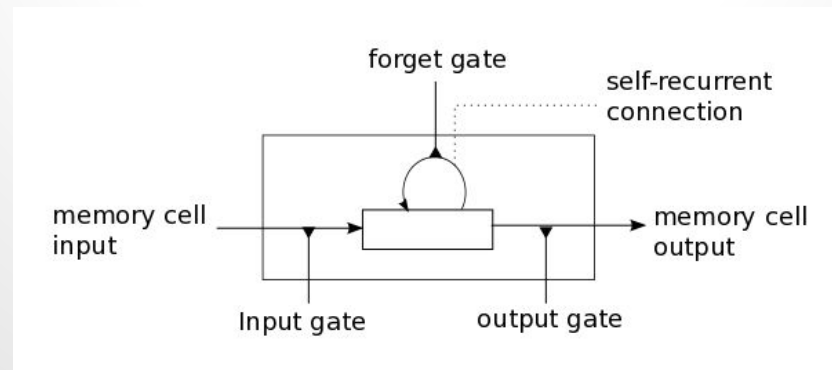
# Going Recursive

Recurrent layer:

Input: current value + previous output



We'll use Long Short Term Memory RNN (LSTM)

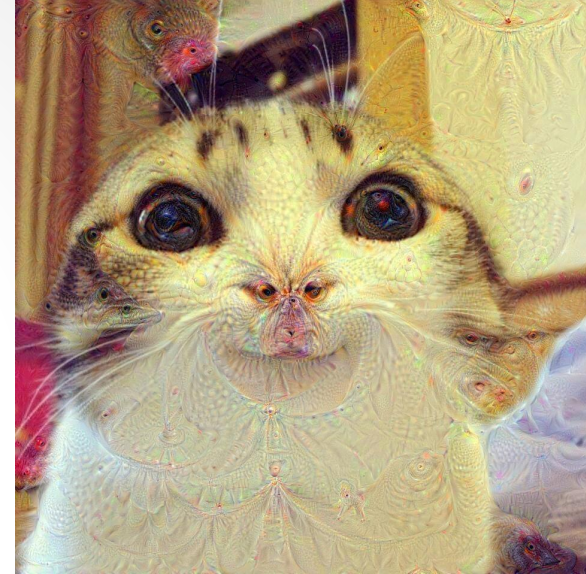


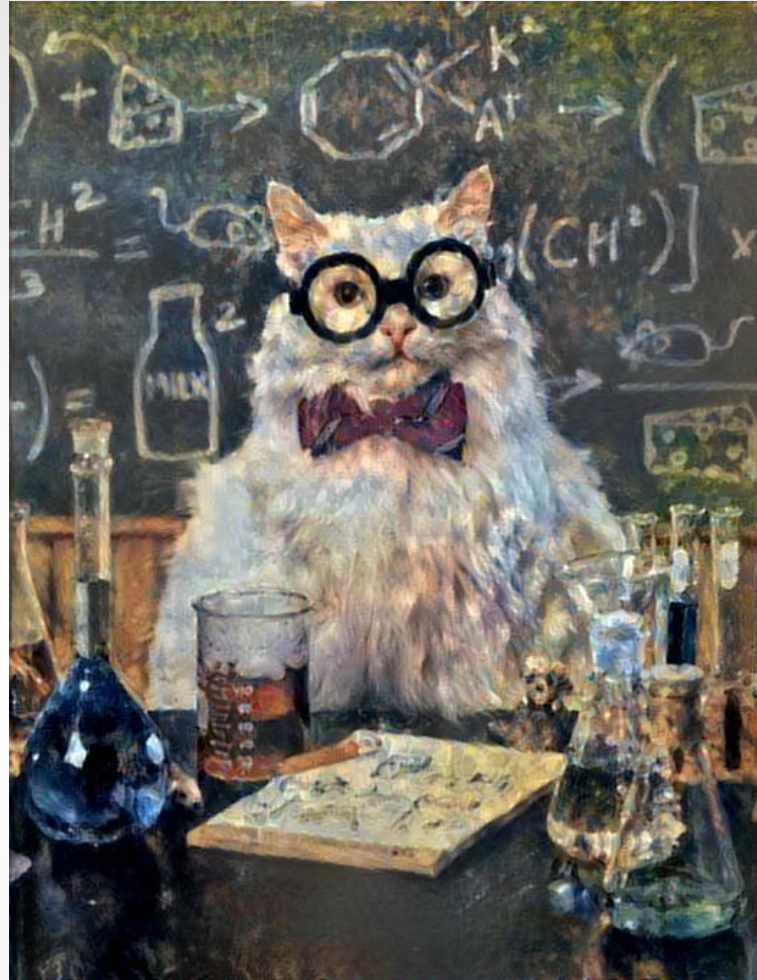


**Yeah we can have sequence inputs !**

# LSTM/RNN success

- English to French translation
- Text generation
- Structured text generation
- Function boundaries in binary files





Can we use LSTM to extract features ?



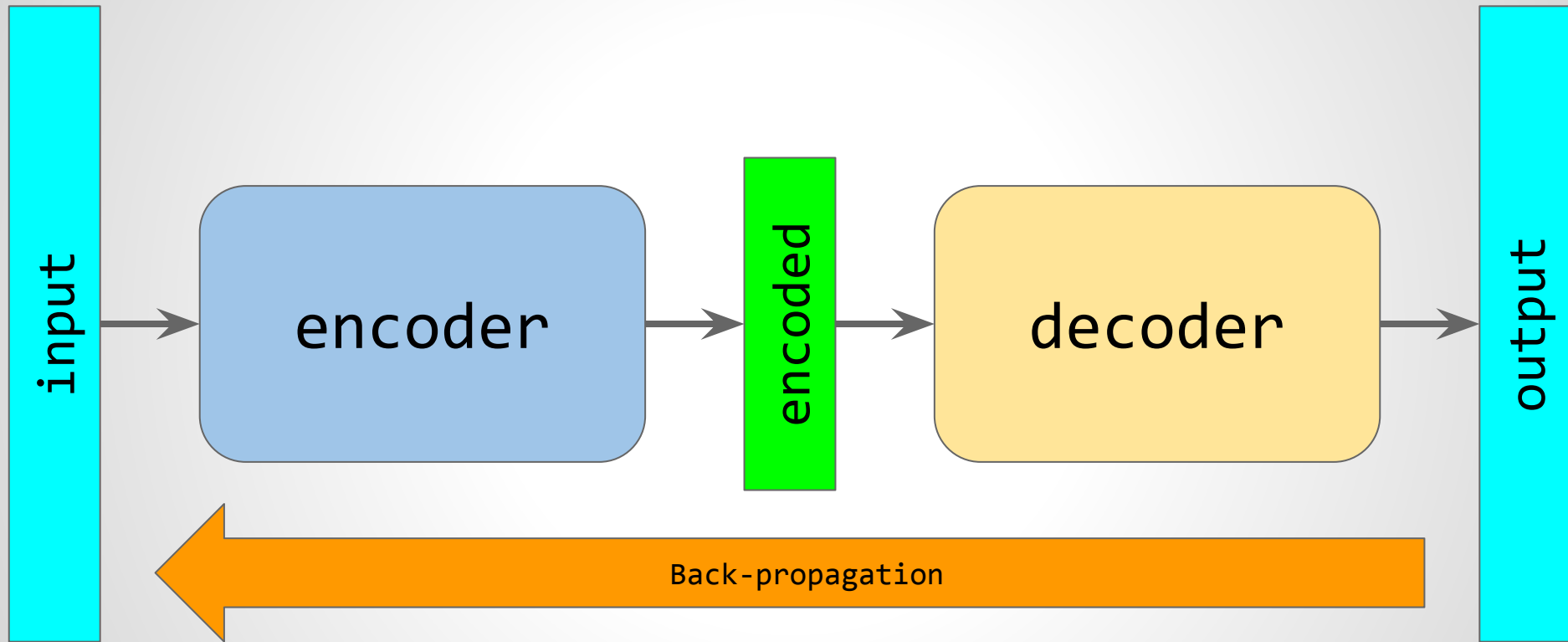
# The problem

- Take arbitrary sequence of bytes
- Extract a finite set of features
- The set should provides good result for distance

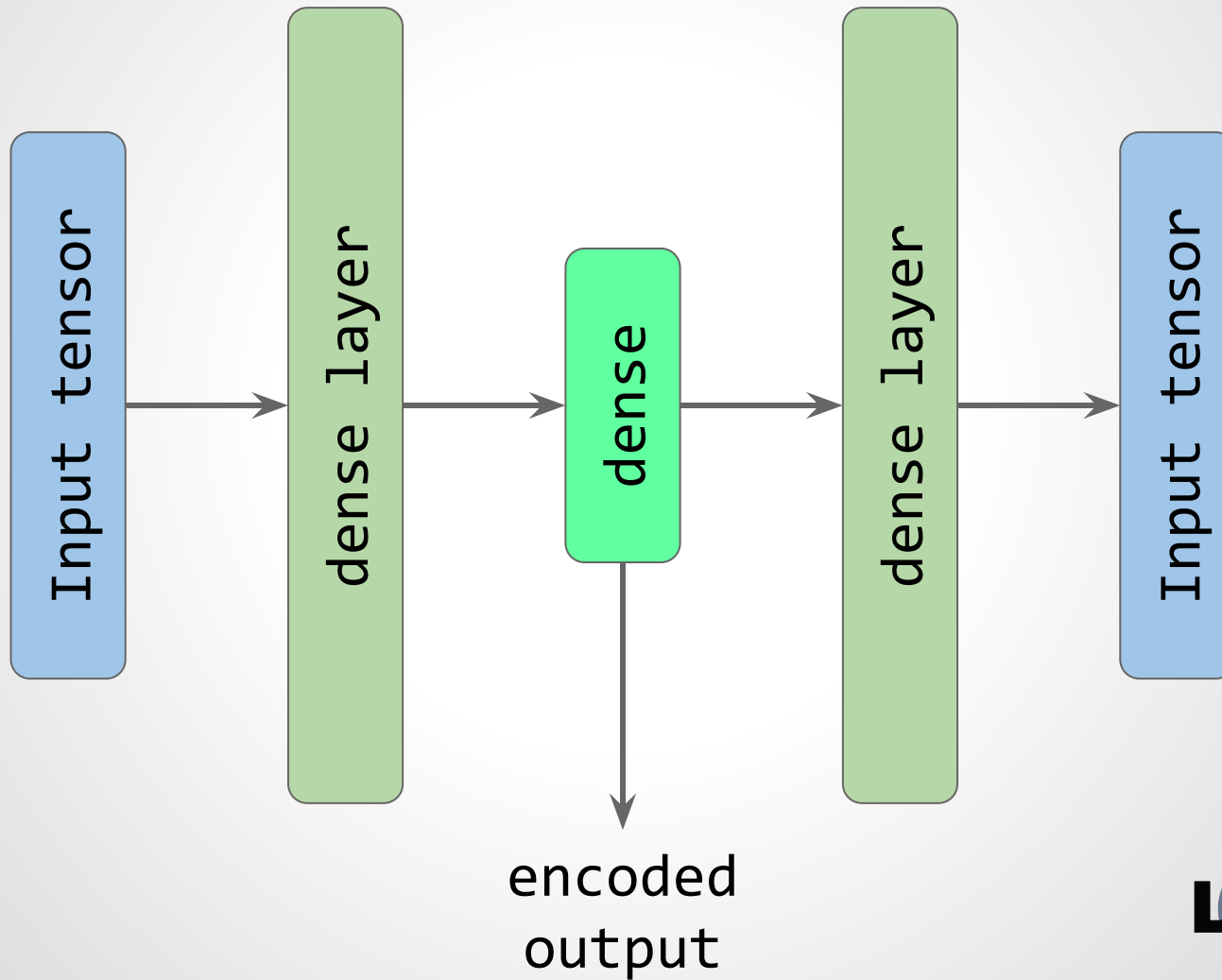


# Auto-encoders

# Auto-what ?

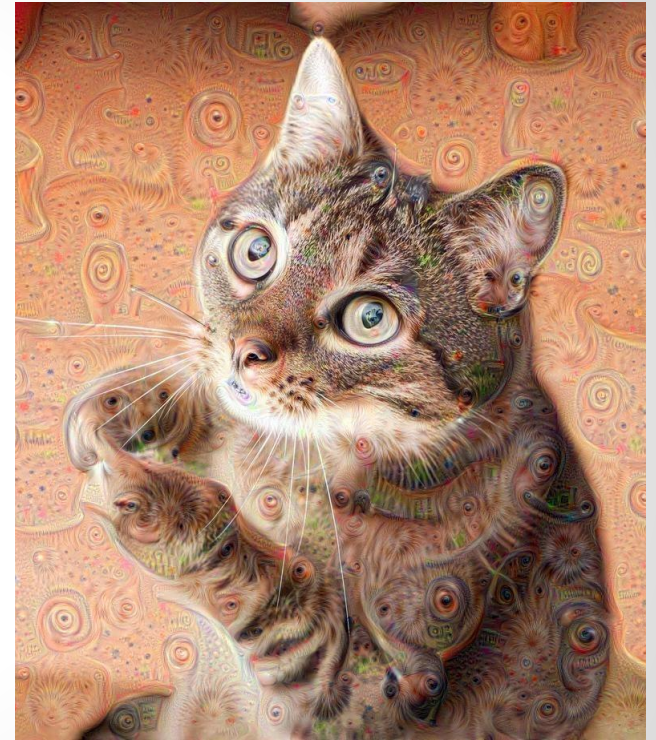


# Basic model



# What for ?

- Pre-training layer
- Data Denoising
- Dimensionality reduction



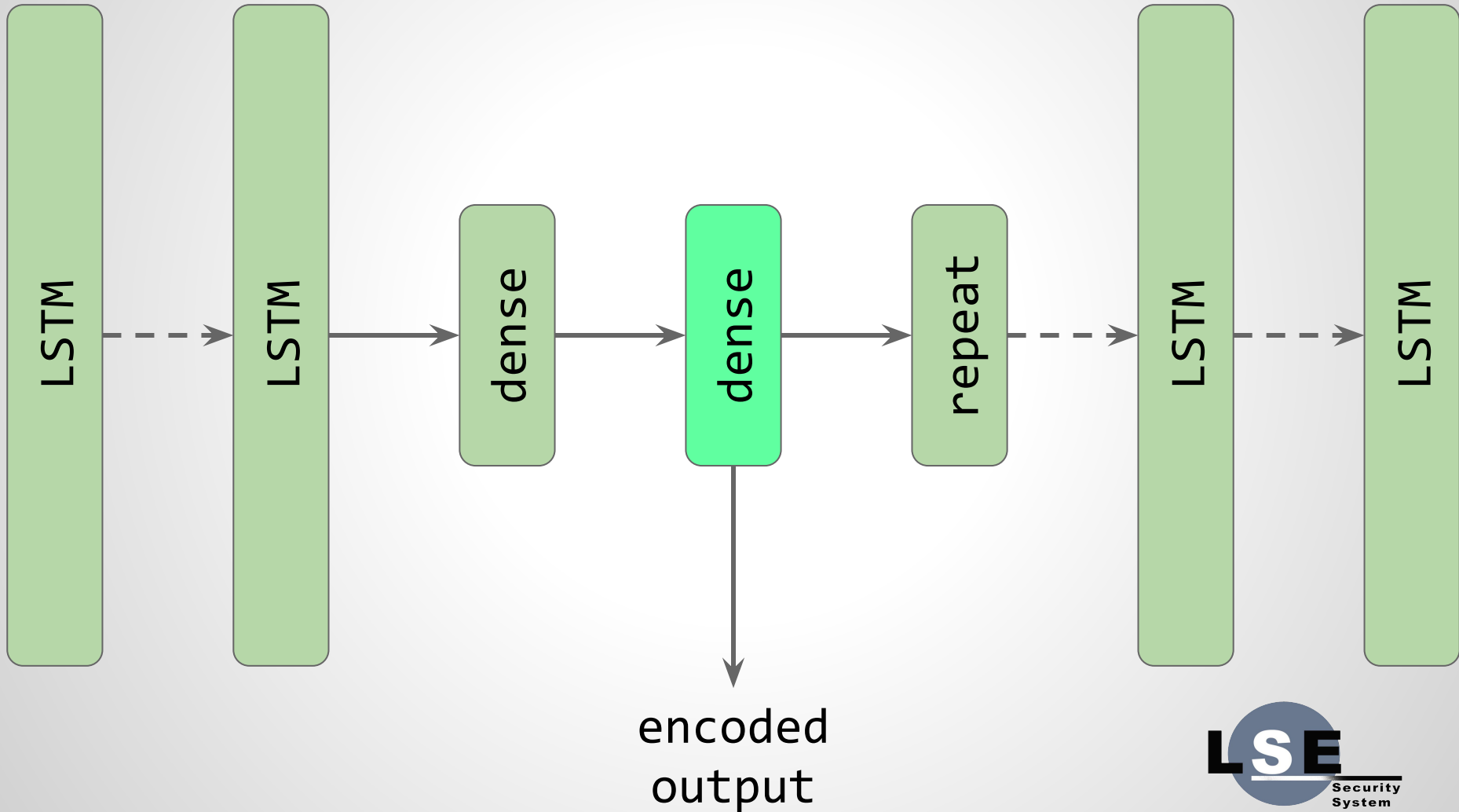


OK, What can we do with that ?

# Sequence Auto-encoders

- Use LSTM layers as input and output
- Add a dense cumulating layer between them
- Rebuild sequence
- We have our feature extractor !

# Binary file auto-encoder

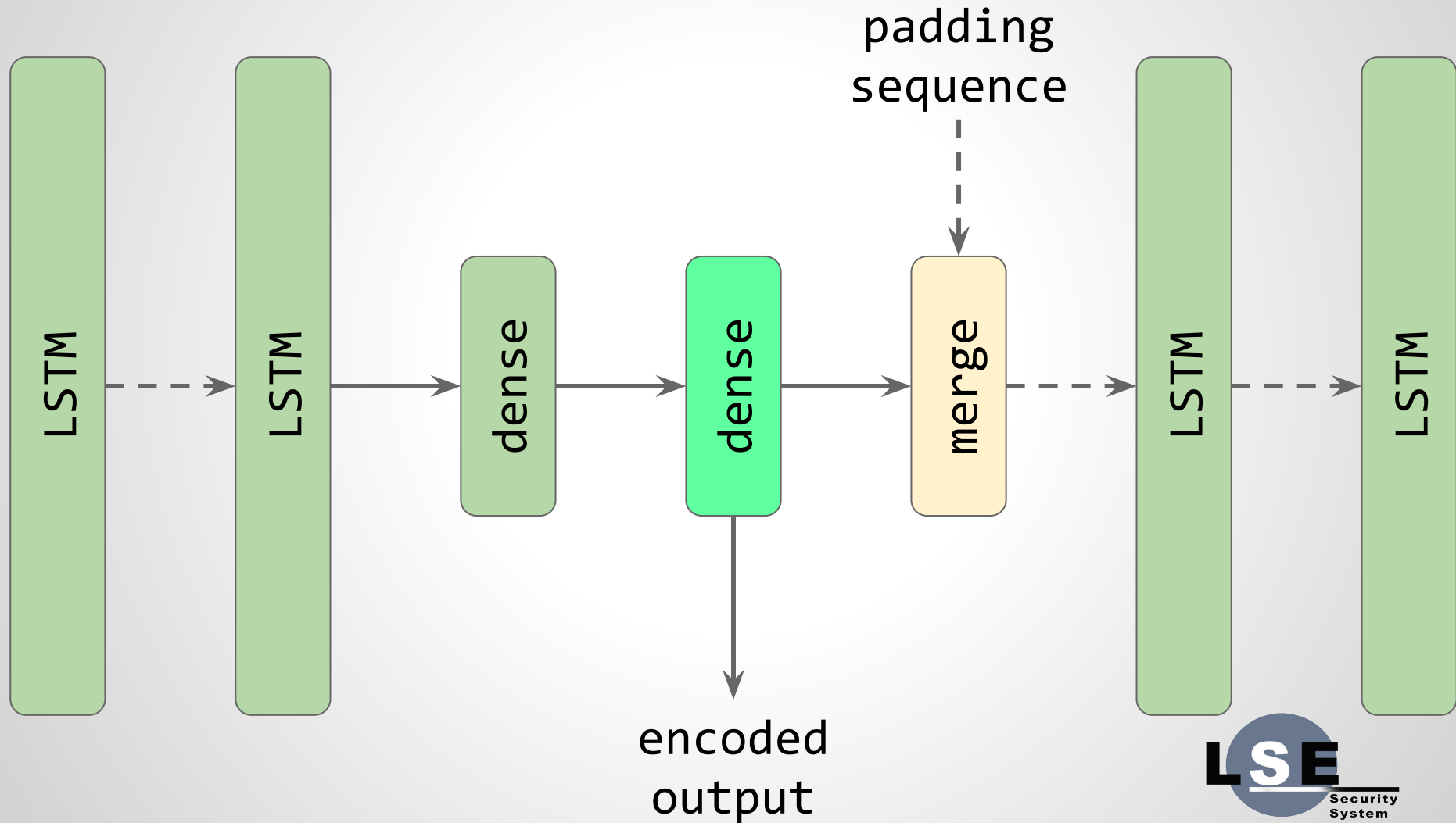




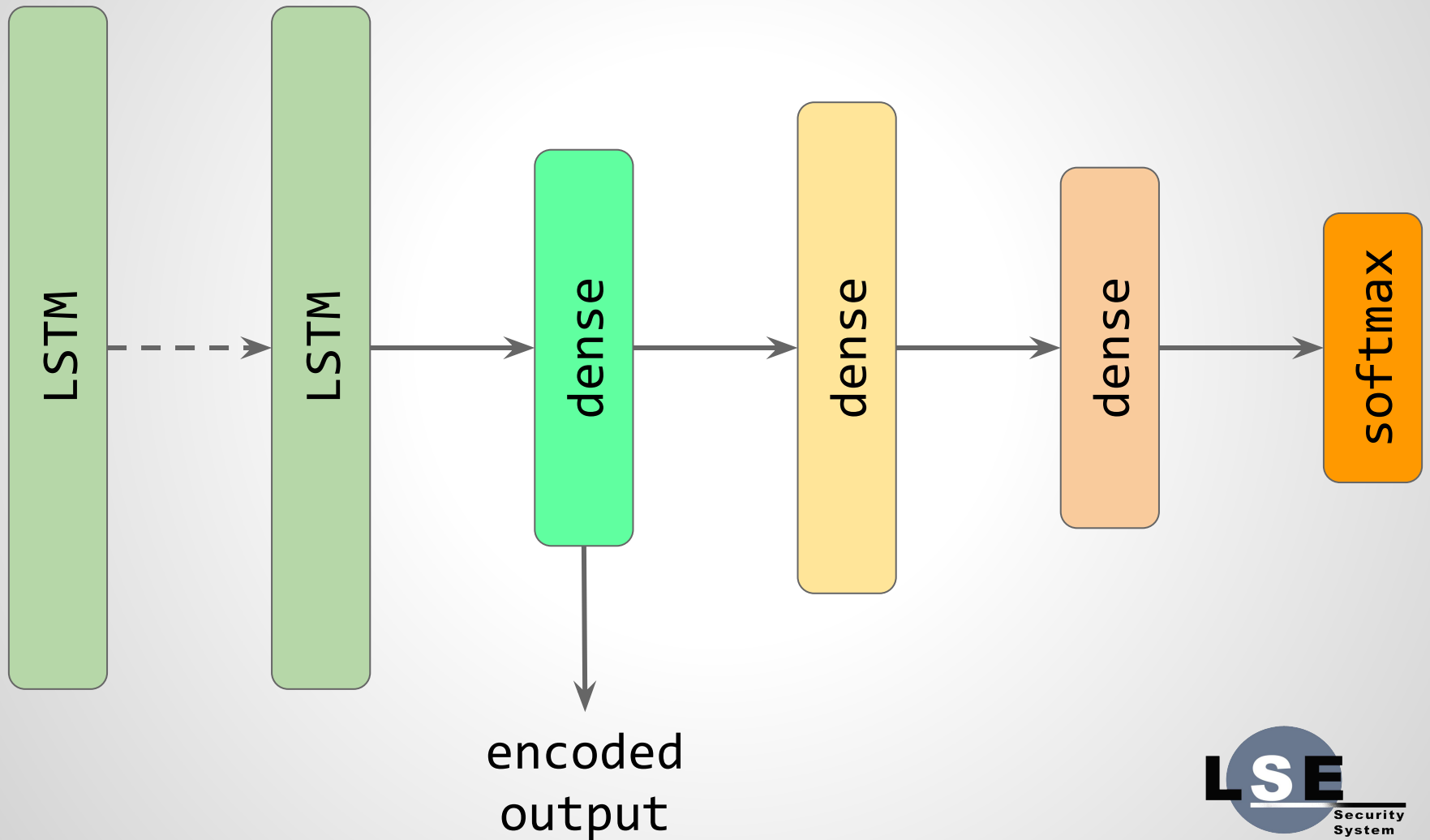
# Sequence length ?

- Repeat layer forces uniform sequence length
- Solutions ?
  - Multiple AE for *each* length
  - Padding all sequences to same length
  - Extra input

# Extra input for repeat



# Classifier for extraction

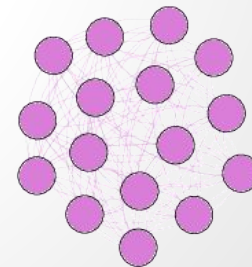




# Results

# Basic C codes

- 5 simple C codes
  - hello world
  - integer square root
  - factorial (iterative)
  - quicksort
  - quick median
- 8 compilers option sets
- Extracted .text section
- Classes: original source code
- 32 features extracted
- graph built using our distance NN



# Students' Code

## Data:

- 18 questions
- 269 students + reference code
- 4164 .text sections extracted from object files
- Labeled by questions

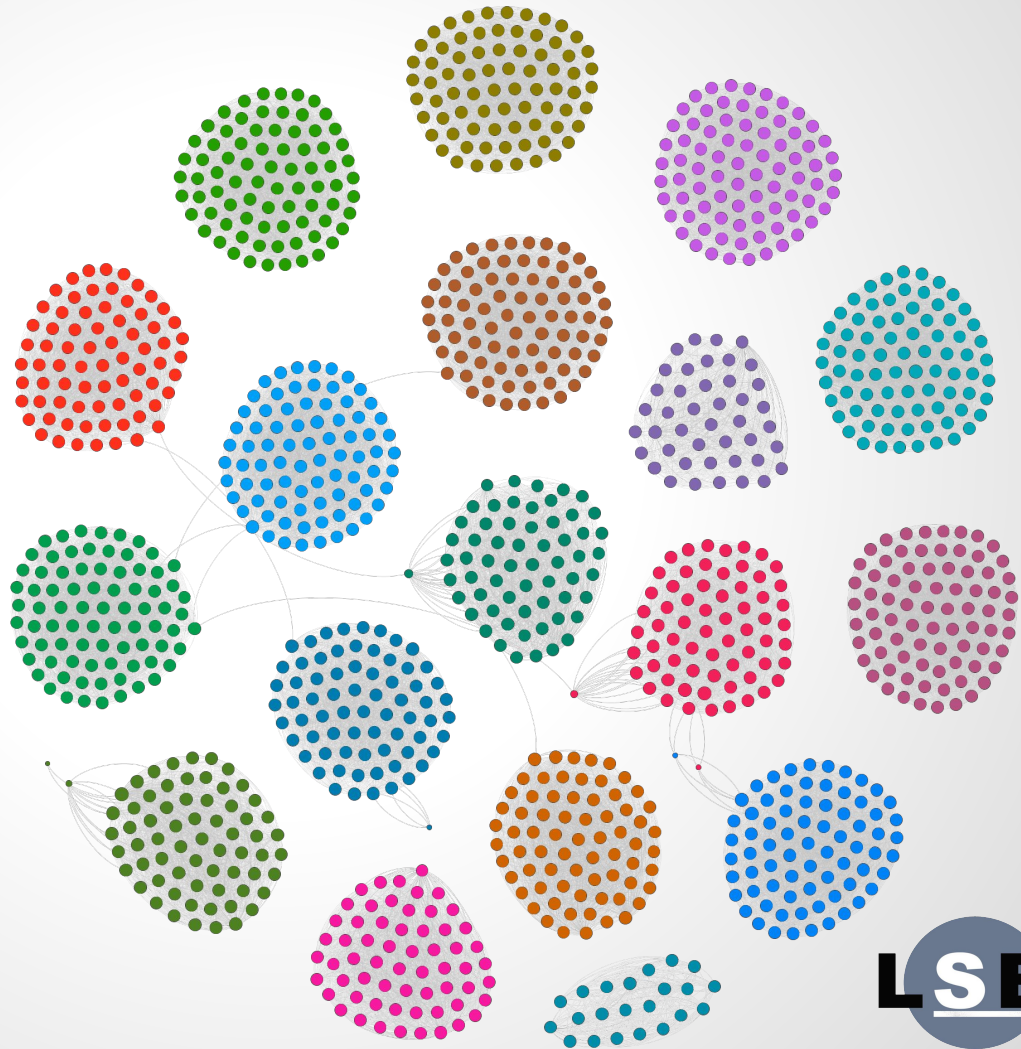
## Classification Results:

4155 correctly classified files

error rate: 0.213 %

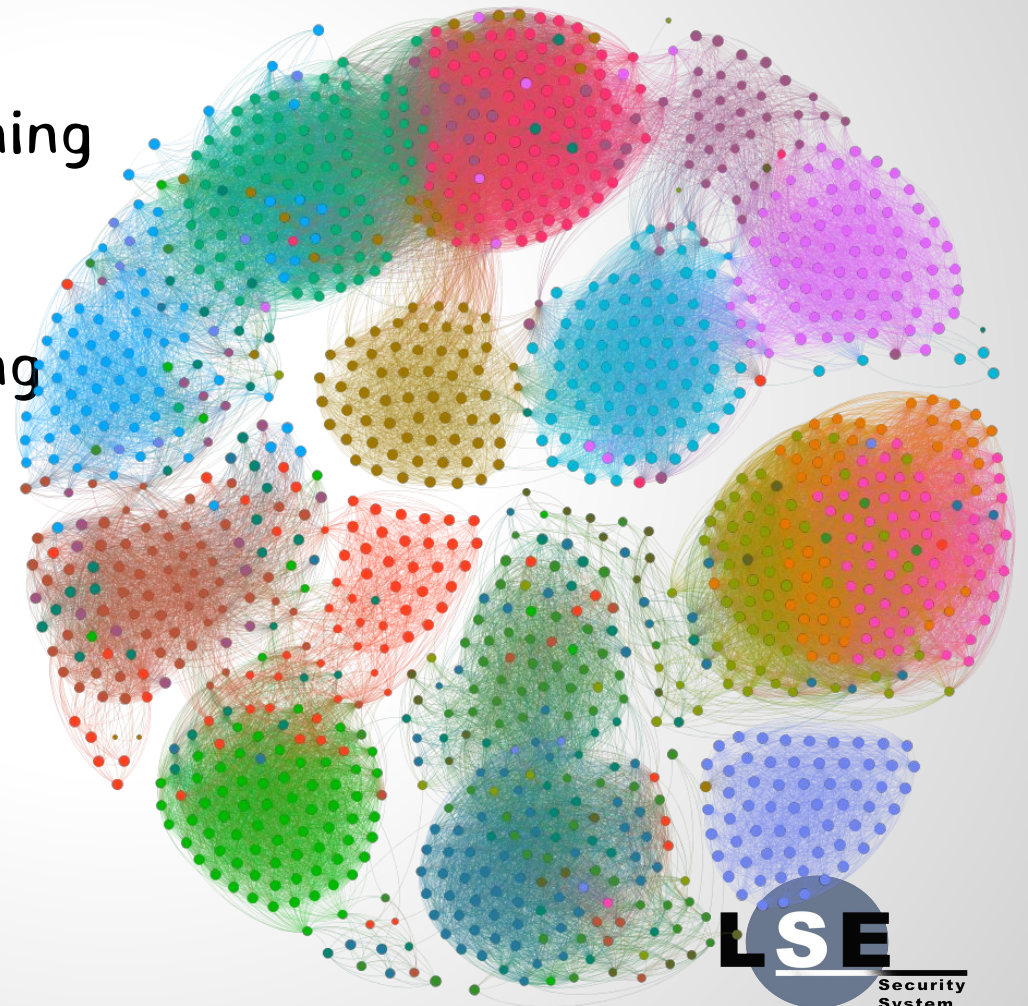
# Students Code Graph

- classifier encoder
- 1239 vertices
- 43534 edges
- Not fully connected
- 18 communities



# Partial Knowledge

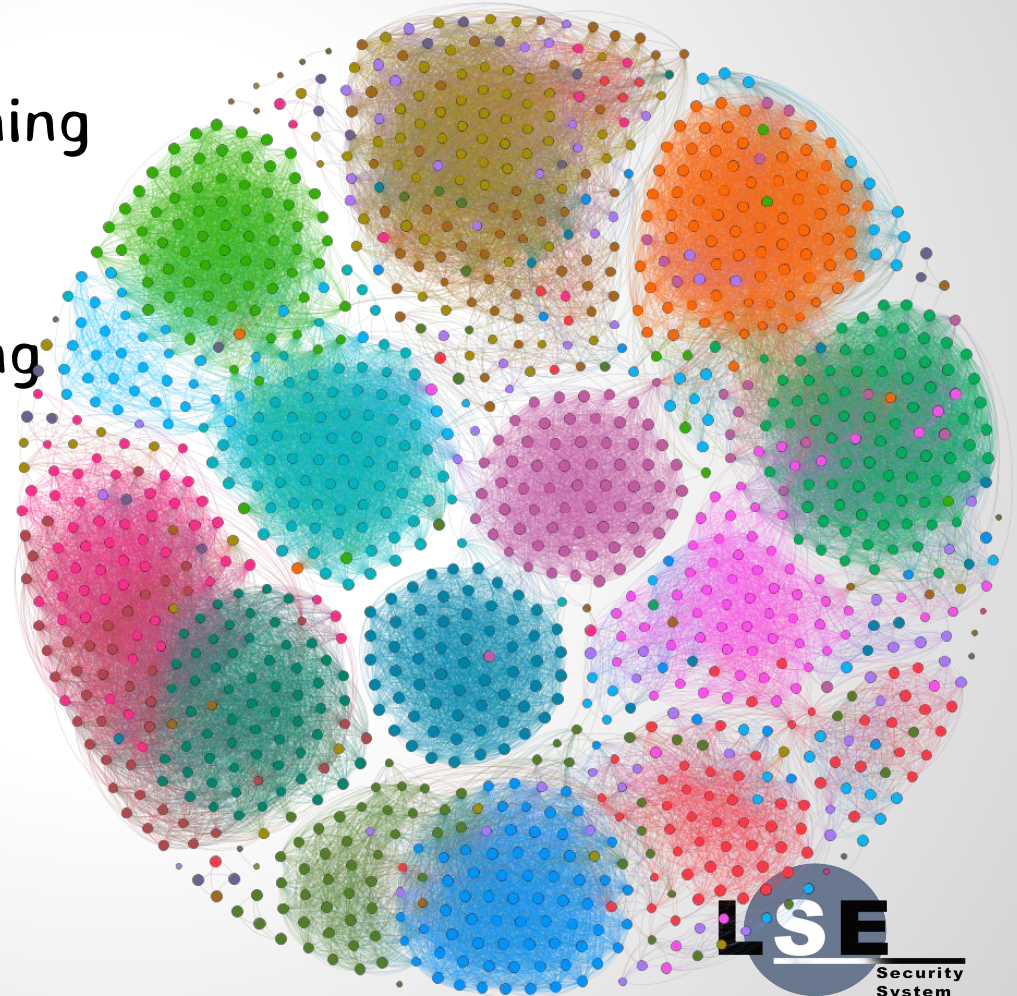
- Auto-encoder
- 14/18 classes for AE training
- Encode all classes
- 20 epochs of edge training





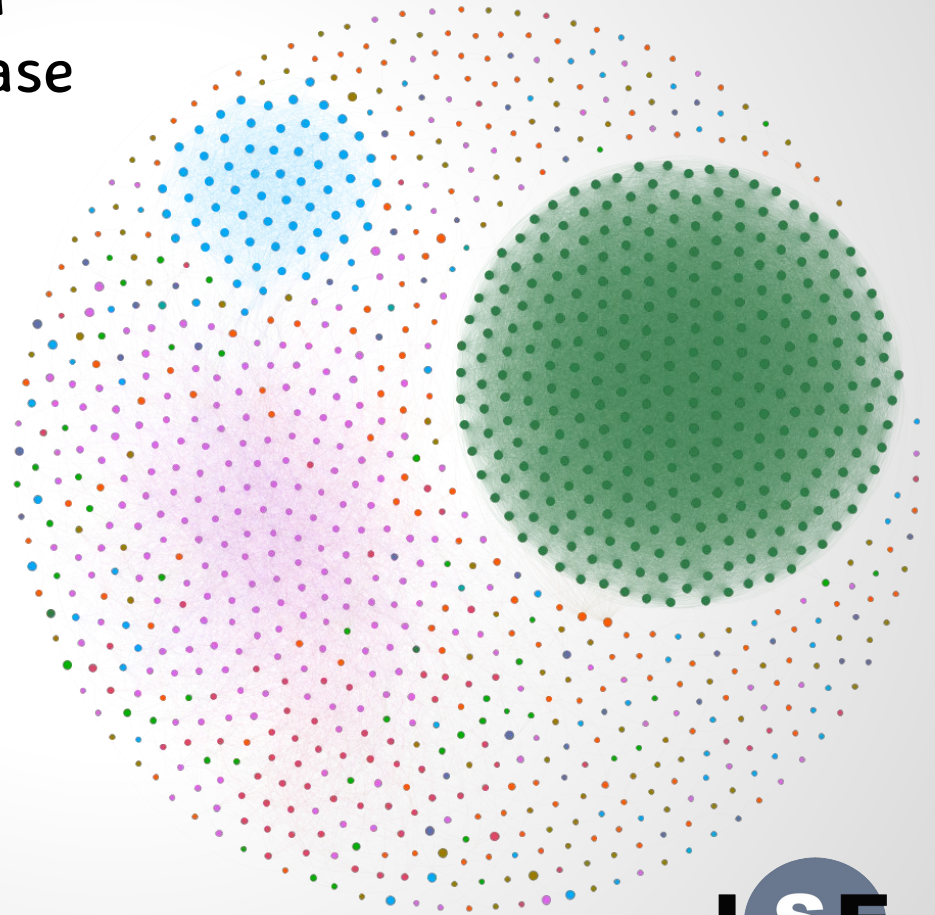
# Partial Knowledge (2)

- Auto-encoder
- 14/18 classes for AE training
- Encode all classes
- 40 epochs of edge training



# And malware ?

- Much longer computation
- Results for 10% of database
- Encoding using classifier
- Not much epoch



# And malware ?

- Much longer computation
- Results for 10% of database
- Encoding using classifier
- Not much epoch
- Another view of the graph





Questions ?